REPORT DOCUMENTATION PAGE

Form Approved
OMB NO. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE 12/13/2003	3. REPORT TYPE AND DATES COVERED Final 01-May –1996 – 30-Jun-2003

4. TITLE AND SUBTITLE
Fielding a New Hybrid Model of Human Learning
Hybrid Learning on the NRL Navigation Task

5. FUNDING NUMBERS
N00014-96-1-0538

6. AUTHOR(S)
Devika Subramanian

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rice University, Computer Science , MS 132
P. O. Box 1892
Houston, Texas

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research Regional Office San Diego
4520 Executive Drive, Suite 300
San Diego CA 92121-3019

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy or decision, unless so designated by other documentation.

12 a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12 b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

The central question addressed by the project is how humans learn complex visualmotor tasks. Can we construct a model of human learning of such tasks based purely on visualmotor performance data? We answer this question in the a_rmative. From a large sequential corpus of visualmotor data gathered from human subjects during learning, we track the evolution of control policies as subjects make the transition from being novices to becoming task experts. The visualmotor data is non-stationary; it is characterized by periods of slow evolution punctuated by conceptual shifts in which policies change radically. We have developed algorithms that build and track models of control policies across these conceptual shifts. These models are rich enough to capture individual differences in the task, and are simple enough to learn in real-time. That is, we have developed methods for learning objective models of cognitive activity (instead of relying on subjective verbal reconstructions) by observing the time course of visualmotor performance. These models can be used to shape and speed up the training of human subjects on complex visualmotor tasks with significant strategic components.

14. SUBJECT TERMS

15. NUMBER OF PAGES
28

16. PRICE CODE

17. SECURITY CLASSIFICATION OR REPORT	18. SECURITY CLASSIFICATION ON THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Hybrid Learning on the NRL Navigation Task
Final report

Devika Subramanian
Rice University

December 12, 2003

1 Project Summary

The central question addressed by the project is how humans learn complex visualmotor tasks. Can we construct a model of human learning of such tasks based purely on visualmotor performance data? We answer this question in the affirmative. From a large sequential corpus of visualmotor data gathered from human subjects during learning, we track the evolution of control policies as subjects make the transition from being novices to becoming task experts. The visualmotor data is non-stationary; it is characterized by periods of slow evolution punctuated by conceptual shifts in which policies change radically. We have developed algorithms that build and track models of control policies across these conceptual shifts. These models are rich enough to capture individual differences in the task, and are simple enough to learn in real-time. That is, we have developed methods for learning objective models of cognitive activity (instead of relying on subjective verbal reconstructions) by observing the time course of visualmotor performance. These models can be used to shape and speed up the training of human subjects on complex visualmotor tasks with significant strategic components.

1.1 Research question

How do humans learn complex tasks with significant strategic and visualmotor components? Examples of these tasks include submarine navigation (e.g, the NRL Navigation Task)and flight control. The tasks are difficult for humans to learn because they require the coordinated acquisition of a strategy (e.g., an evasive maneuver) and the skills to implement it (e.g., a visualmotor servo-loop). Current training methods for such tasks consist of subjects interacting with computer simulations without real-time guidance, followed by an assessment of skills learned. Training systems have no techniques for differentiating between learners who have difficulties formulating high-level strategies from those who simply cannot implement them in their visualmotor system. Further there are no ways to adapt the training protocol for learners based on their specific learning difficulties. Conventional techniques from cognitive psychology (particularly the use of verbal protocols) are not very helpful for these tasks, because humans are unable to access or articulate cognitive processes involved in such learning. The problem of modeling human learning on such tasks is open in cognitive science.

Our hypothesis is that useful, personalized models of human learning can be constructed in real-time by gathering and analyzing data on visualmotor activity in subjects learning the task.

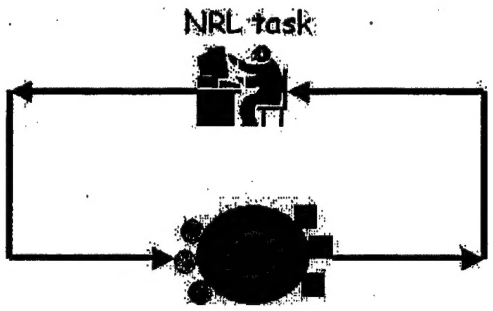


Figure 1: The fundamental research question studied is: can we track cognitive activity during learning by looking over the shoulder of a human subject and unobtrusively recording all visualmotor performance data? Our research has demonstrated that we can build agents that construct models of learning based purely on visualmotor performance data on complex visualmotor tasks with significant strategic components.

Such data includes moment to moment recordings of the visual information presented to the learner, their eyetracker readings, as well as action or motor choices made (by recording joystick motions). Our goal is to build individual models of human learning by analyzing and fusing diverse sources of *low-level* visualmotor performance information. This is especially important in tasks noted for considerable variation among individuals in learning performance.

1.2 Approach

There are two basic approaches to modeling human learning. In the first approach, one posits a general cognitive architecture for learning, independent of the task. Examples of such architectures are SOAR [11] and Epic [9]. One uses data gathered from a subject learning the task to instantiate (and often modify) such an architecture. The approach has been successfully applied on a range of cognitive tasks, and some simple visualmotor tasks such as typewriting. Instantiating a general-purpose learning architecture requires significant effort, expertise and time because there are many free parameters in these models. The approach is not scalable for tasks such as the NRL Navigation task, where there is significant individual variation and there is need for diagnostic models that are built in real-time or close to real-time.

The second approach, which is the one we adopt here, is task-directed. These models are learned without human intervention directly from visualmotor performance data. Figure 2 illustrates the essence of our approach for modeling human learning on the NRL Navigation task. By a model of human learning performance on the task, we mean a representation of a function f from perceptual history (visualmotor history) and time to actions (motor action choices). Since the model evolves with time, we have time as an explicit parameter of the model. Our goal is to abstract the input-output behavior of the human learner, and analyze the evolution of policies for choosing actions, as training proceeds. There are some constraints on the level of abstraction we can adopt: the models need to be detailed enough to pinpoint problems in a subject's learning (e.g., whether strategy formation or skill refinement is incomplete/incorrect); yet be coarse enough (have as few free parameters as possible) to be unambiguously built from the available learning performance data. Other than that, we are completely free to let the task constraints dictate the best model for

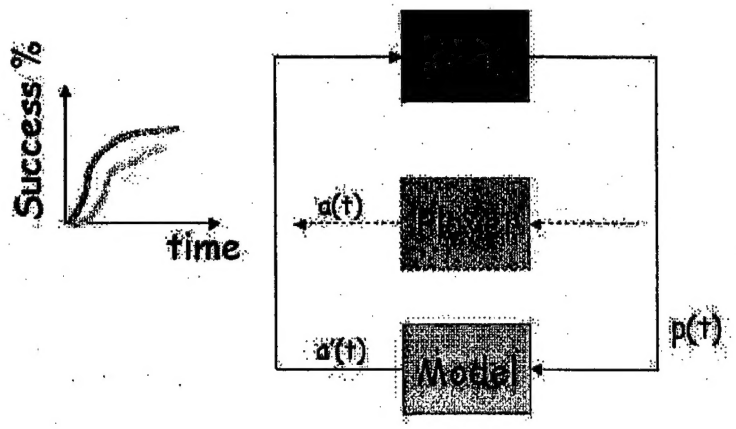


Figure 2: The goal of the modeling is to be able to generate and explain human learning performance on the Navigation task. Replacing the human in the task interaction loop with a model of the human must yield substantively, the same learning behavior (i.e., the same learning curve, as depicted on the left).

explaining and generating human learning performance.

Our task-directed approach to cognitive modeling takes the low level visualmotor data as the ground truth and uses algorithms from information theory, machine learning and data mining, to induce a *compression* of the data. We find a compact representation of the low level data in the form of a policy, mapping visualmotor history to motor action choices. This approach has the advantage that cognitive modeling constructs arise endogenously from the data, rather than being stipulated *a priori*.

How can we measure the quality of our learned models? One criterion, standard in cognitive science, is fit to learning curves. The human's learning curve represents how a task-specific measure of success changes with training time. If the learning curve generated by our model matches the human learning curve, as illustrated in Figure 2, we will say that our model is a good representation of the observed visualmotor performance data.

The rest of this report is organized as follows. Section 2 describes the motivating task: the NRL Navigation task, and the challenges in both learning the task as well as modeling the learning process. The visualmotor performance data is non-stationary and high-dimensional, making the policy extraction task very challenging. In Section 3, we describe our first attempts at learning policies from the visualmotor data [4, 5, 6]. We manually segmented the visualmotor data stream to identify nearly stationary sequences of trials. We then used decision tree learning algorithms to extract policies from the stationary segments. While decision trees prove useful in summarizing learning performance, we do not obtain good fits to human learning curves. To better understand the nature of policies for this task, we built a near-optimal player for it. The player reveals key distinctions that need to be made in the interpretation of sensory information. The strategy adopted by the near-optimal player gives us a baseline for evaluating policies extracted from human performance data. In Section 4, we develop a new, task-specific hybrid model based on the partitions of the sensor space adopted by the near-optimal player [17]. It is a mixture model consisting of action distributions associated with certain sensor space classes, and a hidden Markov model to

capture sequential aspects of human play. While the model gives us high-level characterizations of the strategy employed by learners; it unfortunately, does not provide good fits to human learning curves. In Section 5, we explain why models based on abstractions of sensor spaces cannot meet the stringent criteria of fit to learning curves. We then propose the use of instance-based models to represent mappings from perceptual history to action choices [18]. We develop algorithms that partition the visualmotor data stream into nearly stationary segments using KL-derivatives of the instance-based models. These derivatives are computed by fast randomized sampling algorithms, and they determine when two successive stochastic action policy distributions are significantly "different". The KL-derivative analysis sheds new light on how humans learn strategies for the Navigation task. We discover that humans adopt and discard strategies in a very discontinuous way. Successful learners of the Navigation task display a very characteristic KL-derivative profile. This is a significant finding because we can detect subjects with difficulties in strategy formulation fairly early in the training protocol. We model action selection using an extension to locally weighted regression [2] called *biased dimension elimination*. We experimentally demonstrate the power of our method in coping with the high dimensionality of the performance data. Instance-based models are rich enough to capture individual differences in the task, and are simple enough to learn in real-time. They provide remarkable fits to the human learning curves (see the animations in http://www.cs.rice.edu/ devika/ONR/animations.html). In Section 6, we provide a summary of our work on getting machines to learn the task under the same conditions as humans. We design reinforcement learning algorithms that achieve significantly higher levels of competence than humans. The implications of this finding for human training are discussed at the end of Section 6. In Section 7, we describe student training under the auspices of the grant and conclude with a summary of the impact of the results on the problem of understanding human learning on complex tasks.

2 The NRL Navigation task

The NRL Navigation task requires piloting an underwater vehicle through a field of mines guided by a small suite of sonar, range, bearing and fuel sensors. Sensor information is presented via an instrument panel that is updated in real-time (see Figure 3). The sensors are noisy. Decisions about motion of the vehicle (speed and turn) are communicated via a joystick interface. The task objective is to rendezvous with a stationary target before exhausting fuel and without hitting the mines. The mines may be stationary or drifting. A trial or episode begins with the vehicle being randomly placed on one side of a mine field and ends with one of three possible outcomes: the vehicle reaches the target, hits a mine, or exhausts its fuel. Reinforcement, in the form of a scalar reward dependent on the outcome, is received at the end of each episode.

Since the world is presented via sensors that are inadequate to guarantee complete state identification, the Navigation task is an instance of a partially observable Markov decision process. Fortunately, we can convert it to a fully observable Markov decision process by state augmentation. The state space explodes to 10^{16} states. The augmented state space is very irregular and has no known symmetries. The set of allowed actions in each state is also large; there are 153 possible actions turn/speed combinations in each state. The action space, like the state space, does not decompose naturally. In particular, speeds and turns cannot be learned independently.

There are four major sources of complexity in the Navigation task from the cognitive perspective.

1. *Need for rapid decision making with partial information*: This is one of the chief sources of

Figure 3: The instrument panel for the NRL Navigation Task. There is a bearing sensor, a fuel gauge, a range sensor and seven sonars giving a 140 degree forward field of view. The goal is to pilot an underwater vehicle through a field of mines to a rendezvous point while avoiding mines and without running out of fuel.

complexity of the task. Action decisions have to made in real-time on the basis of incomplete and possibly incorrect information; the environment does not wait for the decision-maker.

2. *Need for competent visualmotor coordination*: The task requires subjects to be comfortable with the use of a joystick and have reasonably good hand-eye coordination. Our training protocols allow subjects to have several initial interactions with the task to become used to the joystick.

3. *Limited binary feedback*: The environment provides very limited feedback; in particular, binary feedback is given at the end of a long sequence of moves. This makes credit assignment very difficult. Subjects have to debug their strategy choice as well as their specific move choices with a single bit of information about the success or failure of a sequence of 200 moves!

4. *Tightly coupled action space*: The action space is two-dimensional. Subjects have to make moment-to-moment choices of speed and turn. Decisions about speed and turn are tightly coupled, and subjects have to learn this dependence in a real-time decision making context with very little feedback.

Together, they make the task challenging for our human subjects; one out of every three of our subjects never acquires the task with our current training protocols. To visualize what is hard about the task, imagine being blindfolded in an unknown room with drifting obstacles, with a noisy sonar for detecting obstacles, and a talking compass. The goal is to get to an exit out of the room within a specific time deadline. In addition, collision with an obstacle terminates the game.

The Navigation task is a natural fit for reinforcement learning because it is a Markov decision problem. An obvious question is whether reinforcement learning can acquire the task within the same constraints as human subjects. Surprisingly, many of the factors that make the task difficult for humans, also make it difficult for machine learners. This is in spite of the fact that machine learners are not handicapped by visualmotor coordination constraints that humans face. The sources of complexity for machine learners are:

1. *Enormous, irregular state space*: The state space of size 10^{16} is a great challenge for reinforcement learners because this particular state space has no known symmetries. Without an appropriate progress measure to guide reinforcement learning, the best that the machine learner can achieve is 3% success on the task after 100,000 episodes of training. This is to be contrasted with our best human subjects, who acquire the task at the 80% level after 1000 episodes.

2. *Large action space*: Most of the work in the reinforcement learning literature concerns itself with environments in which the number of actions associated with a state is less than or equal to 4. The complexity of learning is exponential in the number of actions. To see this, note that the number of actions at each state is the branching factor of that state. Since we consider move sequences of up to 200 in length, we have a search space (distinct from the state space) that is 153^{200} in size!

3. *Limited binary feedback*: Reinforcement learning techniques based on temporal difference [22, 20] propagate credit backward from the goal state. When move sequences from the initial state are up to 200 in length, the number of training episodes needed to determine appropriate moves at the start state can be very very large indeed.

These considerations make a straightforward implementation of a state-of-the-art reinforcement learning algorithm[21] ineffective for the Navigation task. The Navigation task is thus a challenging one for both human subjects as well as for machine learners.

2.1 Data for computational modeling of human learning

Five subjects ran the Navigation task with a configuration of 60 mines, small mine drift, and low sensor noise.[1] An Applied Systems Laboratories (ASL) Model 4000 eyetracker was placed on the head of each subject. The gauge sizes and the visual distances between gauges were sufficiently large to enable the eyetracker to distinguish subjects' focus in almost all cases. A joystick, custom-made by Thrustmaster, Incorporated, was used to input the subject's choice of turn and speed.

Subjects ran consecutive episodes during an hour. The number of episodes per hour varied from around 60 to 160. Each episode varied from a few to 200 time steps (action decisions). All subjects ran for five one-hour daily sessions. At the beginning of the first session, they were told they had to navigate through a minefield to get to a target location and were instructed on how to operate the joystick. Between episodes, the experimenter occasionally asked them to verbalize what they were thinking and learning.

Data was collected on three different media:

1. *visualmotor performance traces* of sequential snapshots of every set of sensor readings and actions taken, along with success/failure feedback at the end of each episode. We have megabytes of trace data: time-indexed sequences of sensor vector and action vector pairs denoting action choices made by subjects during the entire course of training.

2. *fixation files* of every visual fixation.

[1] Five undergraduates at San Diego State University participated in this experiment.

3. *videotape recording* of the instrument panel seen by the subjects on the computer screen, along with a white square denoting the eyetracker's recording of the subject's visual focus of attention, and all *verbal utterances* of the subject. Verbal data gathered from our subjects was not as detailed as we would have liked. It appears that performing the task does interfere with the ability to verbalize. However, utterances from our subjects provide indicators to key shifts in their conceptualization of the task, which is reflected in subsequent differences in their eyetracker and motor behavior.

A major challenge for the Navigation task is the fact that the detailed data we wish to base our models on is at an extremely low level (motor traces, eyetracker data), and high level cognitive information as captured by the verbal protocols is very sketchy.

3 Preliminary analysis of human learning data

In Figures 4, 5 and 6, we show the learning curves of the five subjects; i.e., how the success percentages, timeout percentages and explosion percentages evolve over the course of training. The success learning curves are remarkably similar for the three subjects who eventually acquired the task. The success curves for the subjects who fail to learn the task are also very similar. The failure curves for successful subjects are also alike; however there are individual differences in failure curves among the subjects who did not acquire the task. This raises hope for building a common computational model for all subjects with a few parameters to account for individual variations.

One of the most striking results from the eyetracker fixation data is confirmation of the focus heuristic in an early model for the task posited by us in [5]. Novice subjects distribute their focus of attention rather randomly among the sensors in the instrument panel. The three subjects who developed expertise at the task eventually converged upon an eyetracker pattern restricted to only the sonar and bearing sensors. When the sonar squares are empty, focus is on the bearing; otherwise, focus is on the sonar. As we shall see, this is an important part of the strategy used by the near-optimal player described in Section 3.

By manually analyzing portions of the visualmotor performance data jointly with the verbal and eyetracker data, we observed a significant common thread that runs through all the human subject data. Subjects go through periods of relatively stable performance, punctuated by substantial improvements in performance. This trend is visible in the success learning curves of our subjects. Further examination of the data around these sudden performance improvements reveals that the leaps are associated with radical shifts in conceptualization of the task. These are manifested as shifts in perceptual patterns, which are then followed by shifts in action strategies.

We provide an example of such an analysis for Subject 5, who eventually became an expert at the task. Our subject showed suggestive evidence for her shifts in conceptualization, perception and action strategies well before she verbalized them conclusively. Shifts in the her strategy occurred gradually and unevenly, but once cemented they corresponded to a leap in performance. During session 2, around episode 45, Subject 5 first verbalizes the shift as a hypothesis by stating "only the middle sonar can kill me." By this, the subject means that she can safely ignore all sonar squares other than the middle one. At this point, the eyetracker pattern shifts from attention on all gauges to attention on only the bearing and sonar gauges. When looking at the sonar, attention is more closely clustered near the middle square, as seen in 50-episode fixation and transition summaries. By episode 67, the subject states that her hypothesis is confirmed, and a change in action strategy occurs. In particular, Subject 5's pre-shift strategy is forward motion and fairly

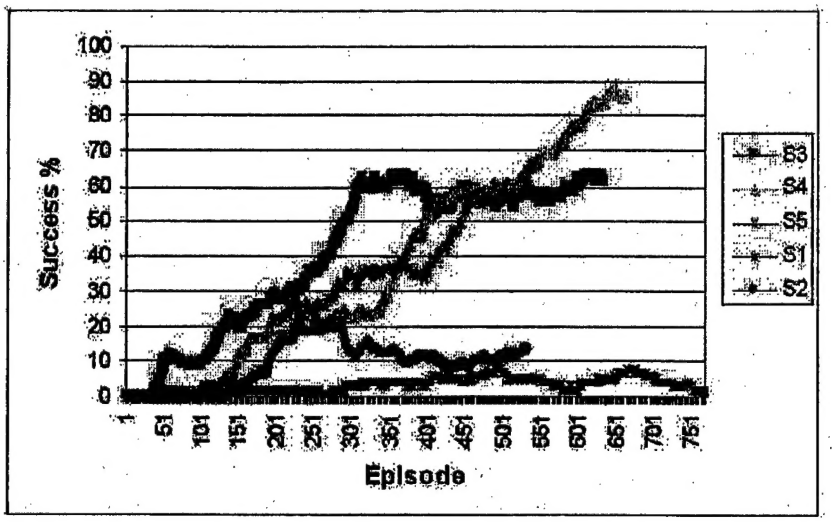


Figure 4: The evolution of success percentages on the Navigation task as a function of training for five subjects.

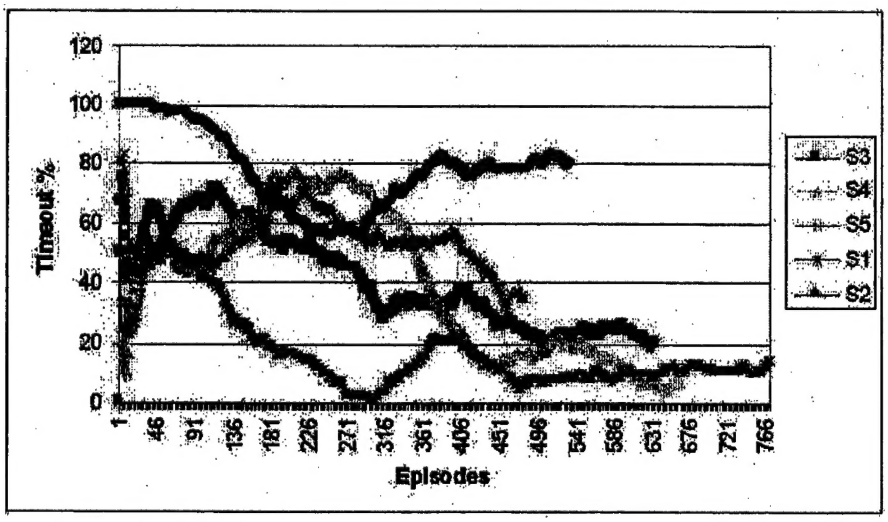


Figure 5: The evolution of timeout percentages on the Navigation task as a function of training for five subjects.

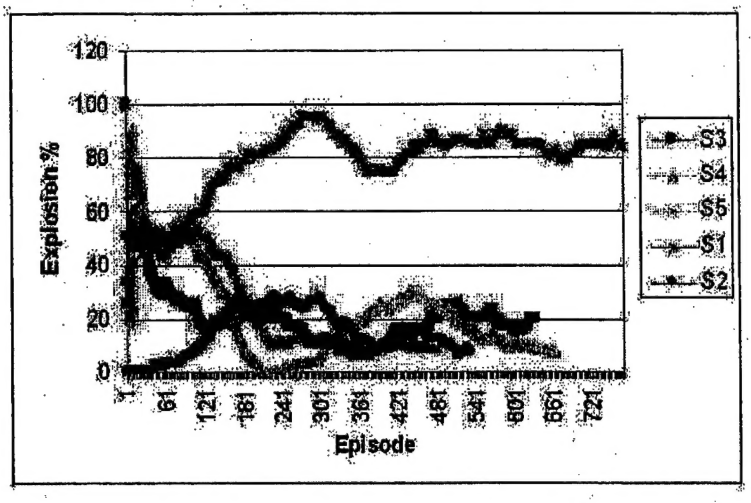


Figure 6: The evolution of explosion percentages on the Navigation task as a function of training for five subjects.

random turn decisions. The post-shift strategy consists of slowing down when she gets closer to the mines, "sweeping" left and right in an attempt to see the direction with least obstruction, then proceeding in that direction. She keeps the bearing straighter toward the target. Figure 7 shows how Subject 5's action probability distributions changed. It also shows the accompanying improvement in performance.[2] It is very interesting to note that the performance improvement is *exclusively* along the dimension of reduced explosions. This is consistent with Subject 5's stated philosophy that "Timeouts are less bad than explosions." We use the decision tree learning algorithm C4.5 [14] to model Subject 5 before and after the shift. The results are in Figure 7. Note that although the magnitudes produced by the model only coarsely approximate those produced by the subject, the trends are captured. For example, both model and subject increase the number of full stops and reduce the number of their explosions after the conceptual shift. A more complete description of this initial modeling effort is in [6].

Our preliminary modeling gave us valuable insight into the ways in which conceptualizations of our subject shift with time, and how conceptual shifts are implemented as new visualmotor strategies. The analysis required manually coordinating the various data sources to find the points during training that corresponded to such shifts. It raised two important open questions: (1) how to automatically find inflection points in data corresponding to strategy shifts, (2) how to improve the quantitative fit of the models to the subject's learning behavior. It was clear that we needed to use richer models to represent the time varying function policy function learned from our subjects.

Since the space of possible sensor configurations is 10^{16}, it seemed clear to us that any representation of the policy function f from sensor readings and time to actions, must reduce the sensor space by finding equivalence classes where action choices are invariant. However, it is not immediately apparent what the right abstractions and discretizations of the sensor space are. To remedy

[2]Based on empirical data, episodes 48-66 are selected for pre-shift, and episodes 67-82 for post-shift.

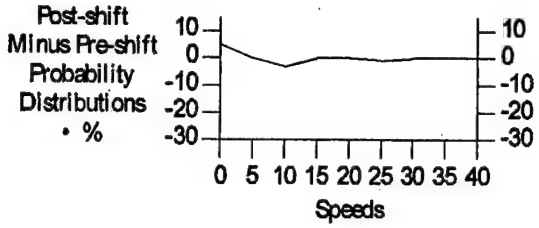


Figure 3: S5's speed differences.

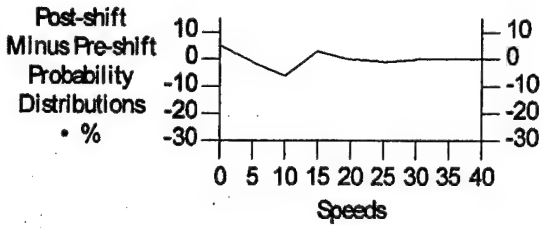


Figure 4: S5 model's speed differences.

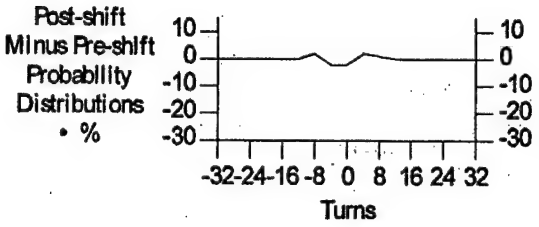


Figure 5: S5's turn differences.

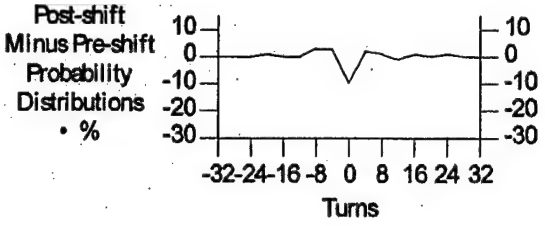


Figure 6: S5 model's turn differences.

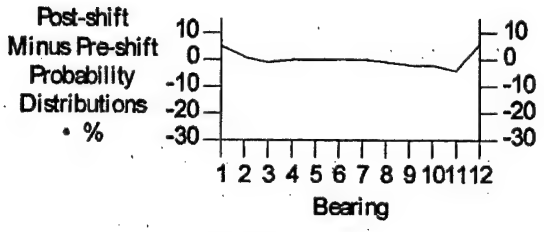


Figure 7: S5's bearing differences.

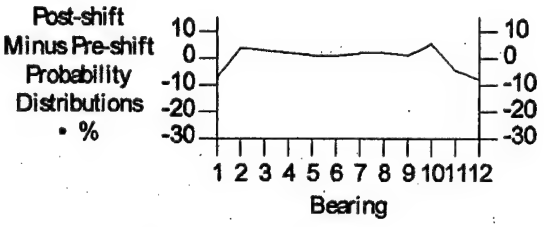


Figure 8: S5 model's bearing differences.

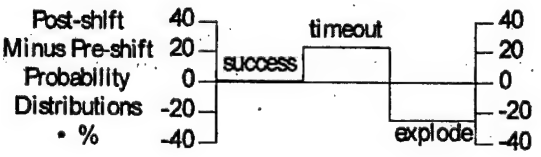


Figure 9: S5's performance differences.

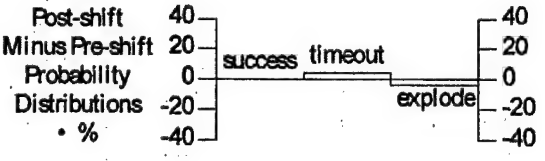


Figure 10: S5 model's performance differences.

Figure 7: Subject 5's learning behavior before and after a conceptual shift. The figures on the right are from the model of Subject 5 obtained by function fitting Subject 5's motor traces.

this situation, we invested effort in designing a computational player for the task, which would play as close as possible to a 100% success rate. Such a player would make the right distinctions in the enormous space of sensor configurations, and we could use it as a basis for evaluating a human subject's strategy.

3.1 A near-optimal policy for the Navigation task

A near-optimal policy for the task is deterministic and is shown in Table 1. It must be emphasized that *discovering this solution was not easy!* It took several months of work with a machine learning system (described in Section 6) to arrive at this policy. There are three key properties of the near-optimal policy.

1. *task decomposition*: the policy decomposes the overall goal into the subgoals of *avoid-mine* and *seek-goal*, a decomposition which appears universal among our human subjects. However, the solutions to the sub-goals are tightly coupled and this is difficult for humans to learn.

2. *dependence between turn and speed choices*: Turning at zero (or close to zero) speeds is essential for success on this task. In addition, turning consistently in one direction while trying to find gaps in the minefield, is crucial.

3. *appropriate discretizations*: the near-optimal policy discretizes the sonar values that range from 0 to 220 into a binary distinction of near/far with the threshold set at 50. The discretizations for the other sensors are described below. This discretization is needed to learn the near-optimal policy quickly. In effect, it defines equivalence classes in the combinatorial state space defined by the raw sensor values.

The near-optimal policy partitions the sensor configuration space into three mutually exclusive and collectively exhaustive components. The first part handles action choice for the portion of the sensor space where a sonar in the direction of the goal is clear. The second part handles the cases when at least one sonar not in the goal direction is clear. The third part handles situations when all sonars are blocked. A sonar is clear if its value is greater than 50, otherwise it is blocked. Goal direction, or bearing, is discretized into three regions: straight ahead (bearing value of 12), to the left (bearing > 6 and < 12) and to the right (bearing value ≤ 6). For mine densities of 60 units (in fact between 10 and 60 units), this policy succeeds at least 99.7% of the time. This is why we believe that the policy is near optimal for the task. This performance has not been matched by our best human subjects. Our experiments in machine learning of the NRL Navigation task, described in Section 6 explain what is needed to lift human performance to this level.

Part 1 of the policy makes the smallest turn in the goal direction. The speed is selected to be 20 (half speed) unless the goal is straight ahead in which case it is 40 (full speed). Turns are thus executed at half speed. Part 2 of the policy determines action when sonars in the goal direction are blocked, but there are other clear sonars. This part makes the smallest in-place turn (with a speed of zero) until the middle sonar is clear. Sonars are polled from the center outward, and the first clear sonar closest to the middle, determines the turn. This portion of the policy ignores goal direction, seeking instead to avoid mines by turning in place. When there are no clear sonars, the third part of the policy determines the action chosen. If the previous action was a turn, it continues with it, so that it can turn consistently in one direction as opposed to flipping back and forth between turns to the left and to the right. A turn sequence is initiated by Part 3 of the policy

Part 1: Seek goal	If the sonar in the direction of the goal is clear follow it at half speed, unless it is straight ahead, then travel at full speed.
Part 2: Avoid mine	Turn in place in the direction of the first clear sonar counted from the middle outward.
Part 3: Gap finder	If the last turn was nonzero, turn again by that amount, else initiate a turn by summing the sonars to the left and right, and turning in the direction of the lower sum.

Table 1: The three-part near-optimal policy for the NRL Navigation Task

Controller	**Success %**	**Behavior**
Original	99.7	
Original - Part 3	79.9	When sonars are blocked, oscillates back and forth in place. Loses by timing out.
Original - Part 2	98.3	Amazingly effective without this part. Loses by timing out.
Original - Part 1	7.3	Since it ignores bearing, never gets to goal. Loses by timing out.
Part 1 in isolation	50.1	Very aggressive goal seeker. Always loses by blowing up.

Table 2: Performance of ablated versions of the near-optimal policy in Figure 1 for mine density of 60. All win percentages are compiled over 10,000 episodes.

if one isn't already ongoing. It selects a mild turn to the right if the sonars to the right are clearer than the sonars to the left.

We performed ablation experiments to test the importance of each of these parts in the policy. We systematically excised each part and ran the policy for 10,000 episodes. The win percentages and remarks on behavior of the policy are in Table 2. Surprisingly, part 2 of the policy contributes little to the overall performance which is determined by parts 1 and 3 almost exclusively. With no avoidance strategy (provided by parts 2 and 3) and powered purely by a goal-seeking component, the policy does rather well, winning 50% of the time.

So what can we learn from the near-optimal policy for the Navigation task? The decomposition of the task into the *avoid-mine* and the *seek-goal* components is made explicit by the policy's structure. Part 1 purely focuses on goal seeking and chooses actions to minimize the time to reach the goal, without worrying about mine avoidance. Parts 2 and 3 focus on solving the *avoid-mine* goal in the context of the *seek-goal* objective. The Navigation task cannot be decomposed into completely de-coupled subgoals and this is one of the main sources of complexity of this task. Any action that takes the vehicle away from a mine would qualify as an optimal action if we considered the *avoid-mine* goal independently. However, Part 2 looks for the smallest deviation from the as-the-crow-flies path to the goal. If such a path is unavailable, Part 3 rotates the vehicle in place consistently in one direction and as little as possible, till a gap opens up and the the policy of Part

1 or Part 2 applies. The policy implemented by Parts 2 and 3 comprises a solution to *avoid-goal* that is constrained by the requirements of *seek-goal* to reach the target as expeditiously as possible.

The structure of the optimal policy supports the following partition of the enormous state and action space for this task. The number of states in the finest discretization of the task is $22^7 * 15 * 12 * 200 * 17$ which is roughly 10^{16}. Sonars have 22 values and there are seven sonars, range has 15 values, bearing has 12 values, the length of each episode does not exceed 200 steps, and there are 17 possible previous turns. For each of these states there are $17 * 9 = 153$ actions to choose from. The near-optimal policy collapses many of the distinctions made by such a fine discretization. The effective number of states considered by Parts 1 and 2 of of the policy is $2^7 * 3$ which is 384. This is because both parts consider the values of seven sonars which is each discretized into clear and blocked, and three values for bearing. These 384 states are really equivalence classes over $(22-5)^7 * 15 * 12 * 200 * 17 \approx 10^{14}$ states in the state space. Part 3 examines the previous turn discretized into two values: zero, and non-zero, and thus deals with an effective state space of two!

The analysis of the near-optimal policy gives us a novel way of examining the voluminous visualmotor data collected from humans. In the next section, we describe a new methodology for studying visualmotor data from humans by evaluating deviations from this near-optimal policy. The approach directly yields lesson plans for training humans to rectify current deficits in their strategy choice.

4 Building hybrid models of human learning

To help with the analysis of the low level visualmotor data with existing algorithms, we adopt the near-optimal policy as a baseline. This policy provides the necessary discretization of the sensor configuration space for model construction. A further advantage of using the near-optimal policy as a baseline against which to compare human subject performance is that deviations from the optimal can be the basis for directed training of subjects. A potential disadvantage is that some humans may not adopt anything close to the conceptualization needed for near-optimal performance.

Armed with the discretizations supplied by the near-optimal policy, we derived general accounts of what our successful subjects are learning. A preliminary analysis of the motor data from the three successful subjects reveals that they learn

1. to follow the as-the-crow-flies strategy in the direction of the goal in states in Part 1.
2. to slow down significantly when turning.
3. to turn minimally to avoid mines in states in Part 2.
4. to turn in place consistently to find gaps in minefield in Part 3.

The near-optimal policy provides a justifiable basis for segmenting the motor sequence data into Part 1, Part 2 and Part 3 stages. To fit Part 1 and Part 2 data, we use action probability distributions: i.e., we estimate the probability $P_1(a)$ of taking action a in sensor configurations that belong to Part 1, and $P_2(a)$, for sensor configurations belonging to Part 2. Recall that Part 1 states are those in which sonars in the direction of the goal are clear, and Part 2 states are those in which some sonar not in the direction of the goal is clear. Part 3 states are those in which no sonar is clear. To fit Part 3 behavior, we use hidden Markov models (HMM) because the policy used by our subjects is inherently sequential. The overall structure of the hybrid model that we

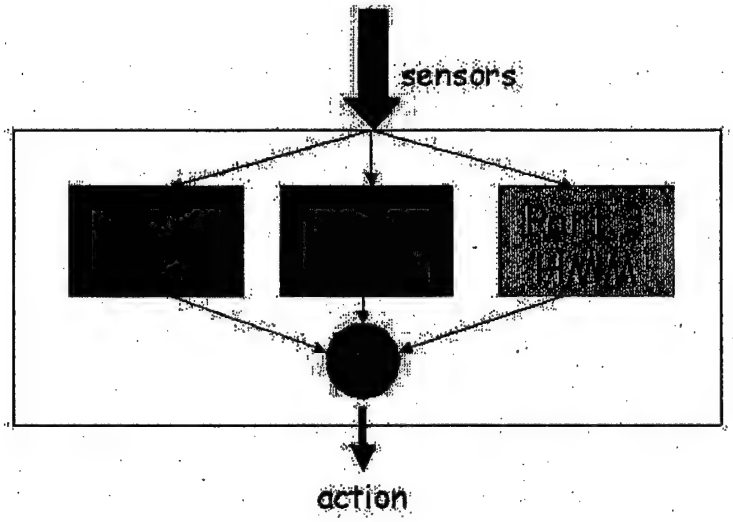
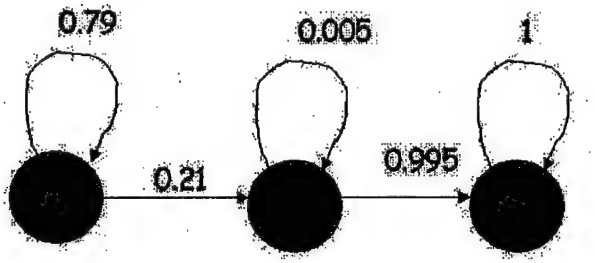


Figure 8: The structure of our hybrid model for the Navigation task.

construct from the subject data is shown in Figure 8. Note that the structure of the model reflects the task structure. In particular, we use probability distributions to fit the subject behavior on the seek-target subgoal of the task, and a combination of a probability distribution and an HMM to model the solution of the coupled subgoal of avoid-mine of the task. Since we know the near-optimal policy for each of these equivalence classes of states, we can determine the deviation for each subject from it, and tell them either explicitly or implicitly (through tailored exercises) how to improve their behavior on those classes of situations.

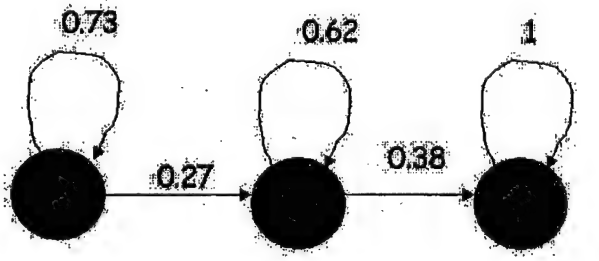
As an example, consider the models we constructed for Subject 5 on her behavior on Part 3 states in Figure 9. Prior to her conceptual shift, her strategy, which we induced from her data using an HMM learning algorithm, is as follows: she pauses at zero speed and turn for a while, and then makes an average of two moves with non-zero speed and turn and finally settles into oscillating left and right at zero speed until time runs out. After her conceptual shift, we acquire the HMM shown in Figure 10. We can directly read off her strategy as: pausing at zero speed and turning for a while, making a left turn at zero speed and then settling into a pattern in which she consistently prefers turning at zero speed to the right. This is fairly close to the near-optimal policy for such states. In fact, with practice we can get her to spend less time in the state labeled 1, and completely eliminate state 2, and in state 3, we can zero out her tendency to pause and increase her probability to turn right. This would form the basis from which lessons will be created to help the subject acquire greater competence at the task.

How good a fit to performance does the model in Figure 8 provide? The results on Subject 5 for Session 2, before and after the conceptual shift are shown in Table 3. The quantative fit to the subject's behavior is superior to the decision tree models we considered in [6]. The superiority of the current model comes from the fact that it closely mirrors the action choice distributions of the subject. In fact, the model *is* a compact representation of these distributions. Such a model meets the criteria of being fine-grained enough to capture strategic regularities in the subject's action



0	0	0.84	0.00	0.36
right	0	0.03	0.003	0.33
left	0	0.05	0.00	0.31
other		0.08	0.997	0.00

Figure 9: A hidden Markov model that generates and explains the behavior of Subject 5 in states where all sonars are blocked, before her conceptual shift.



0	0	0.82	0.05	0.37
right	0	0.024	0.00	0.553
left	0	0.025	0.95	0.07
other		0.131	0.00	0.00

Figure 10: A hidden Markov model that generates and explains the behavior of Subject 5 in states where all sonars are blocked, after her conceptual shift.

Pre-shift	Successes	Explosions	Timeouts	Total episodes
Subject 5	0	12	11	23
Model	0	17	6	23
Post-shift	Successes	Explosions	Timeouts	Total episodes
Subject 5	0	2	13	15
Model	0	4	11	15

Table 3: The behavioural fit of the new hybrid model to Subject 5.

choice and is coarse enough (has very few parameters) to be learnable in real-time. In addition, it has the virtue of being able to provide direct guidance to a teacher as to the design of lesson plans to improve the expertise of the subject.

5 Instance-based models for tracking the evolution of human learning

The hybrid model described in the previous section is an excellent fit to the data in the early stages of learning. However, as the subject's performance improves, the optimal policy no longer is a good filter for the data. All our attempts to manually as well as automatically (by clustering) learn the "right" discretizations of the sensor space made by the subject failed.

We then turned to instance-based models as a paradigm. These models require no abstraction and deal directly with the raw visualmotor performance dara. The visualmotor performance data for NRL Navigation is treated as a two-level time series. At the top level, we have a sequence $\mathbf{E}$ of episodes $e_1, e_2, \ldots e_N$, where each episode itself is a time series of form $\{(p_i, a_i)|i > 0\}$. The sensor configuration vector $p_i \in \mathbf{P}$ ranges over the discrete set $\mathbf{P}$ (10^{16} in size) of all sensor inputs (a 11 component vector: range, bearing, last turn, last speed and seven sonar readings). The motor output $a_i \in \mathbf{A}$ is drawn from 153 turn and speed choices in the set $\mathbf{A}$. We extract stochastic policies of the form $\pi : \mathbf{P} \rightarrow \mathcal{P}(\mathbf{A})$. The choice of stochastic policies is dictated by the fact that there is usually more than one "correct" motor output for a given sensor configuration. We therefore associate a probability distribution over $\mathbf{A}$ for each element in $\mathbf{P}$. Extracting a stochastic policy from the episodic time series above is difficult for several reasons.

1. the high dimensionality of the discrete sets $\mathbf{P}$ and $\mathbf{A}$.

2. non-stationarity of the policy, since it changes with training. Let $\pi_{i,j} : \mathbf{P} \rightarrow \mathcal{P}(\mathbf{A})$ denote the policy extracted from contiguous episodes $e_i, \ldots, e_j, N \geq j > i$ of $\mathbf{E}$. We need to partition $\mathbf{E}$ into n maximal, non-overlapping contiguous segments $[1, i_1], [i_1 + 1, i_2], \ldots, [i_{n-1}, N]$ which span the interval $[1 \ldots N]$ such that the policy is stationary over each segment.

3. non-white noise in the data caused by joystick hysteresis and lapse of a subject attention. Such noise is particularly hard to deal with in the context of non-stationary data, since we need to distinguish the case when there is more than one correct motor output for a given input p, and when one or more of the motor outputs associated with p in the data is incorrect.

5.1 Segmentation by computing policy derivatives

The essential idea for finding stationary segments is to extract stochastic policies from episodes in blocks of size w. We define a distance measure over the space of policies, and compute the distance between policies over two successive blocks. We study how this distance changes with time. We use the standard threshold of the mean plus twice the standard deviation on the change in policy distance between successive blocks to identify blocks where there is a significant change in the action policy. The choice of w is dictated by the two competing factors. Making w as large as possible allows us to construct policies that span $\mathbf{P}$ better, which as we shall see, reduces the complexity of calculating the distance between two policies: Making w as small as possible allows us to pinpoint the location of the policy shift more accurately. We experimentally determine the best value of w for our data to be 20.

More formally, let $\pi_{i,i+w}$ be the stochastic policy derived from episodes $e_i \ldots e_{i+w} \in \mathbf{E}$ and let $\pi_{i+w+1,i+2w}$ be the stochastic policy derived from the next block of w episodes, $e_{i+w+1}, \ldots, e_{2w}$. Suppose we have a distance function Δ_P over the space of all policies. Then the policy derivative at the block of width w that starts at episode i is

$$\delta(i,w) = \frac{\Delta_P(\pi_{i,i+w}, \pi_{i+w+1,i+2w})}{w}$$

To design an appropriate Δ_P, we first fix a representation for the policies. We consider local policy models of the kind created by nearest neighbor methods and locally weighted regression: a lookup table of the form $\{p, \mathcal{P}(\mathbf{A})\}$ where p ranges over the sensors seen in the given block of w episodes, and $\mathcal{P}(\mathbf{A})$ is the distribution of motor outputs associated with p in the same block of w episodes.

For this representation of policies, a natural distance measure is to compute the average distance between action distributions associated with each p that occurs in both policies. We choose KL-divergence of two discrete distributions as a measure of the distance between them. For lookup tables with 10^3 to 10^4 distinct p's, it is impractical to compute the average distance over all p's. To get around it, we compute the average distance over a small number of randomly chosen p's from the policy tables. We experimentally determine that repeatedly and randomly sampling 5% of the p's from each lookup table keeps the variance of the distance measure obtained under 1% of the mean of the distance distribution. In Figure 11 we use this Monte-Carlo sampling technique to rapidly estimate the policy derivative for Subject 3.

There are about five policy derivative peaks higher than the cutoff of the mean plus twice the standard deviation of the distribution of derivative values for the subject on the left of Figure 11. Overlaid on the policy derivative measure is the learning curve of the subject. Note that there are plateaus of performance (on the learning curve) corresponding to the policy derivative peaks. These plateaus represent unchanged performance profiles and we associate them with the use of a fixed policy. Our derivative measure accurately picks out the rising edges of the performance graph between these two plateaus, where the subject's performance undergoes major improvements, which we believe are associated with significant modifications to the action policy. Consistent with predictions from the cognitive science literature, we find that there are only a few shifts in strategy or policy in human subjects learning this task.

We applied the same technique to the visualmotor data obtained from a subject who couldn't learn the task. The policy derivative or KL-derivate profile shows significant cognitive activity, as the subject adopts and discards policies rapidly through the entire training period. We believe that

there is an impedance mismatch between the visalmotor learning and strategy formulation phases for this subject. She formulates strategies too quickly and gives up on them before they are fully implemented through her visualmotor system.

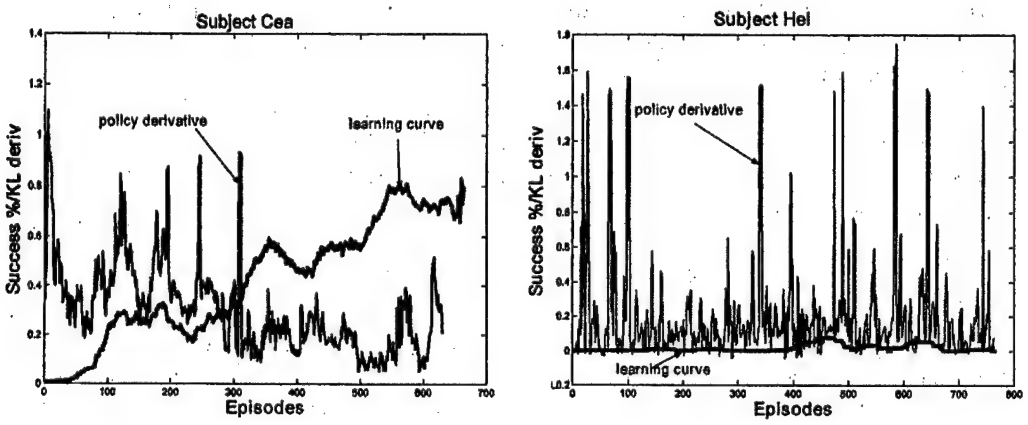


Figure 11: This figure shows the variation in strategy (blue curve) used by two subjects learning the NRL Navigation task, superimposed on their learning curves (red curve). Both subjects train over 600 trials on the task. We measure strategy variation from visualmotor data. We calculate the KL-divergence between distributions of actions chosen by the subject in successive blocks of trials. While the subject on the right successfully acquires the task, the subject on the right fails. The strategy derivative curves for both subjects show abrupt shifts, which correspond to the adoption of significantly different policies for task performance. While the subject on the left shows about 5 abrupt shifts over a 300 trial period with distinct spacing between them, the subject on the right adopts and discards new strategies much more quickly. The shifts occur rapidly; within 10 trials or about 3 minutes in real-time. The learning curve for the subject on the right does not reveal the significant cognitive activity that the visualmotor data stream demonstrates.

5.2 Learning models of control policies

Now that the episodic data is segmented into nearly stationary segments, we learn models that map sensor configurations to distributions over actions using data from each of these segments. The first model we construct is simply a lookup table which contains for each observed sensor vector in the episodes, a distribution of the actions that were taken in response to it. Given a vector p in the lookup table, we output an action drawn from the stored action distribution associated with it. If p is not in the lookup table, we locate its 100 nearest neighbors using the L_2 norm of the difference between the two sensors vectors as the "distance function". To facilitate the search for the 100 nearest neighbors, the perceptual input vectors are stored in a kd-tree. The kd-tree representation of the stochastic policy model is about a megabyte in size, which is a extremely small compared

to the estimated size of $\mathbf{P}$ (10^{16}). We then investigated two methods for computing an action for input vector p using these nearest neighbours:

1. **Weighted Averaging:** An action is computed as the weighted average of the actions taken for each of the nearest neighbours, inversely weighted by the distance to the the vector p.

2. **Locally-weighted regression:** We fit a surface to the neighboring perceptual vectors using a distance weighted regression [2].

Both methods use the same distance function, viz. the L_2 norm of the difference between two sensor vectors, where each dimension is scaled to be in the range $[0, 1]$. The kernel width for locally weighted regression is set to assign a weight of 0.01 to the neighbour that is twice as far away from the query as the nearest neighbour.

We experimentally discovered that locally weighted regression performs worse than weighted average for our data. This is surprising, and contradicts common wisdom about these methods. Closer investigation revealed that the available data to compute nearest neighbors from is extremely sparse (10^4 points in a space of size 10^{16}), causing locally weighted regression to extrapolate, rather than interpolate between neighbors. To fix this problem, we introduce *biased dimension elimination*. The idea is the removal of all those dimensions from neighbouring vectors whose values over the neighbours are all greater or all less than the value of that dimension in p. The regression is then guaranteed to interpolate the action instead of extrapolating it. We conjecture that the procedure might have a cognitive equivalent in that interpolation is more easily performed mentally than extrapolation. When presented with vectors containing biased dimensions (as defined above), the subject will choose to ignore the biased dimensions rather than spend time calculating the correct extrapolation.

5.3 Testing the learned models

To test the performance of our models against that of the subject, we use a standard train/test protocol. Devising a fair protocol for testing prediction accuracies over of a non-stationary data source is extremely difficult. This is because wholesale revision of policies at the strategy shifts need to be taken into account for properly formulating the "train" and "test" sets. Fortunately, by day 5, most of our subjects have no major shifts in policy, so we use a standard cross-validation protocol to test the fit of the models to the learning curve of the human subjects. We divide the data for day 5 into 10 chunks and build a model out of 9 of the chunks and test it on the left out chunk. The process is repeated 10 times and the average performance of the models (measured as number of successful episodes in a given window) is reported. We ran this train/test protocol on all of our subjects.

5.4 The need for richer policy models

It became clear experimentally that the performance of models that calculate action based purely on the current sensor vector do not match human performance at all. This is because stateless models are not rich enough to capture the action choice basis of our subjects. By analyzing the situations where stateless models generated actions inconsistent with our human subjects, we discovered that action choice in situations very close to mines was a function of a prior action history of 4 (the action at time $t-1$, $t-3$, $t-5$ and $t-7$), and the difference between the sensor vectors at time t

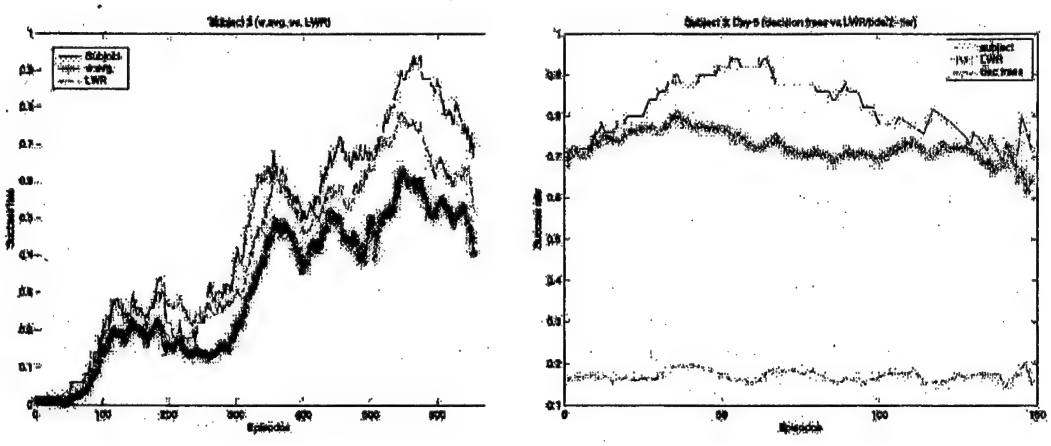


Figure 12: The performance of a two-tier instance-based model using locally weighted regression augmented with biased dimension elimination provides excellent fits to a subject's learning curve. The figure on the right shows the performance of decision trees on the same data (for day 5 of training).

and $t-1$. Thus for a very small fraction $\mathbf{M}$ of the sensor configuration space $\mathbf{P}$, a purely reactive stochastic policy $\pi : \mathbf{P} \rightarrow \mathcal{P}(\mathbf{A})$ is insufficient to generate the right action distributions. Our local models therefore have a two-tier structure. Over all but a small subset $\mathbf{M}$ of $\mathbf{P}$, we learn a reactive stochastic policy that calculates motor output on the basis of the current sensors alone. However, for the subset $\mathbf{M}$ (corresponding to situations which forms less than 2% of the size of $\mathbf{P}$), we capture more state to accurately predict the action choice behavior of our subjects. Each of these policy components is conceptually represented as a lookup table, and physically stored as a kd-tree.

Figures 12 summarizes how well locally weighted regression with biased dimension elimination on the two tier policy representation match human learning curves. It is surprising to see this degree of fit between a simple model learned from the subject's performance data and the subject himself. Note that we are able to capture individual differences in learning with the same model. Our models are simple to construct and they can be built in real time, making them very useful for shaping training of subjects. A comparison of the action distribution associated with a particular sensor vector learned from the subject and that of a machine learned optimal policy can be used to advise subjects about correct actions for those inputs. To further improve the fit to human performance, we need to enrich policy representations and automatically learn the subset $\mathbf{M}$ of inputs for which more history is needed.

We also learned decision tree representations of the policy function $\pi : \mathbf{P} \rightarrow \mathcal{P}(\mathbf{A})$ from the stationary episode segments. The performance of this global technique is shown in Figure 12. The decision trees are very large (with over 10,000 nodes) — they are unable to learn appropriate partitions of the sensor configuration space $\mathbf{P}$. This demonstrates the fact that for our data, local models are better fits than global ones. We believe this holds because of the wide variation in action distributions between the various regions of $\mathbf{P}$. Augmented with a good clustering technique that creates action-equivalent partitions on $\mathbf{P}$, we expect to have better fits to global models such as the ones learned by decision tree.

5.5 Summary of results on instance-based modeling

We have developed methods for real-time tracking of learning in the context of a paradigmatic example of such tasks (the NRL Navigation task) by analysis of the low-level visualmotor data (joystick movements and eyetracker data) gathered during training. From the visualmotor data, we reconstruct the action choice policy used by the human, and track its variation with time over a 5 day training protocol. We discover that the visualmotor data is non-stationary; and that the action policy is characterized by periods of slow evolution, punctuated by radical conceptual shifts in which policies change dramatically. We also discover that successful learners experience between 4-5 such shifts in the first half of the training period. There are no shifts observed in the second half of training, yet performance continues to improve as the strategy gets compiled in the subject's visualmotor loop. Surprisingly, humans who do not learn the task, experience many more (30-40) conceptual shifts over the entire training period, and their policies never stabilize with time. This observation gives us useful measure for evaluating subjects during training. It provides early prediction on whether the subject will successfully acquire the task over the full training period. Our action policy models are detailed enough to capture individual differences in the task, and are simple enough to learn in real-time. We experimentally demonstrate the effectiveness of our modeling techniques by showing the closeness of fit between model and subject performance (see http://www.cs.rice.edu/projects/ONR/animations.html for animations showing subject and model performance).

6 Machine learning the NRL Navigation task

Our primary goal in performing the machine learning experiments was to obtain an optimal policy for the task. We were also interested in answering the following questions.

1. What does it take to get machines to learn the task? In particular, what extra knowledge do we need to provide in order to get reinforcement learning to converge on this task's state space?
2. Can machine learners achieve higher levels of competence than humans on this task?
3. How does the sample complexity of humans compare with that of machine learners? Do machine need more training episodes than humans to achieve the same level of competence on the task?
4. How can we use the results of machine learning to improve human learning?

We have answered the first three questions as part of our work on this grant. With an appropriate discretization to reduce the complexity of learning, a non-deceptive progress function that provides intermediate feedback, and a good credit assignment policy, it is feasible for reinforcement learning to converge to an optimal policy. We have answered the second question in the affirmative: machine learners can and do achieve significantly higher levels of competence than human learners. The sample complexity of machines compares favorably with that of humans.

To make reinforcement learning feasible, we reduced the state space size from 10^{16} to 768. Each sonar was reduced to a binary distinction (1 if its value was greater than 50, and 0 otherwise), and the twelve bearing directions were reduced to six aggregate headings. The number of actions

was discretized to 24. An open problem is the automatic construction of such discretizations from explorations in the original state space.

Careful analysis of the task and experiments with reinforcement learning revealed the design of an appropriate intermediate progress measure to guide learning. Without this progress measure, reinforcement learning reduces to a random walk on a space with branching factor 24 and depth 200. We designed a reward measure designed to bias the random walk into convergence to the optimal policy.

r(s,a,s') is: 0 if s' is a terminal explosion failure state.
2000 if s' is a terminal success state.
1000 if s' is a terminal timeout failure state.
1500 if s is a Part 3 state, and s' is a Part 1 or Part 2 state.
$1000 + 3 * \Delta sum$ where *sum* is the sum of the sonar readings if s and s' are Part 3 states.
$1000 - 2 * \Delta Range + 50 * abs(bearing - 6)$ otherwise. The 2 and 50 are simply the values which maximized performance for the particular scaling used.

We believe that communicating this reward measure in some fashion to our subjects will significantly enhance their learning ability.

We also needed to modify the credit assignment policy used by standard reinforcement learning. Reinforcement learning based on temporal differences penalizes all actions in a sequence that ends in a failure. However, this is not appropriate for an action sequence which leads to hitting a mine. It is only the last action which causes destruction by a mine, so no previous actions should be penalized. This can readily be seen by the fact that in any state the subject could choose to set the speed to zero and avoid hitting a mine. Making this change to the credit assignment policy speeds up reinforcement learning and allows it to converge to a policy to within 10% of the performance of the optimal in about 2000 trials. No human subject achieves this level of competence within these many trials, suggesting ways in which we can boost human performance by making more effective use of each trial.

We studied the possibility of staged learning in the context of the reinforcement learning system. We simplified the task of the learner to be that of learning turns alone, with speed being automatically set by the near-optimal policy. The task of learning optimal turns can be done with a considerable simplification to estimating the sum of rewards, the standard estimation problem for temporal difference learners. In particular, maximizing the local reward shown above, also maximized the global sum of rewards. So a greedy learning method works effectively and is guaranteed to converge to an optimal policy. The performance of the learner acquiring the optimal turn policy is shown in Figure 13. The learning performance of the full learner that acquires turn and speed simultaneously is also shown in this figure. The difference between these two learners suggests a new protocol for training humans. We can train subjects to first learn turns, and then learn turn and speed choices together. We predict that learning the full task will proceed much faster (with fewer training episodes), and will allow subjects to achieve higher levels of competence. This conjecture remains to be tested on human subjects.

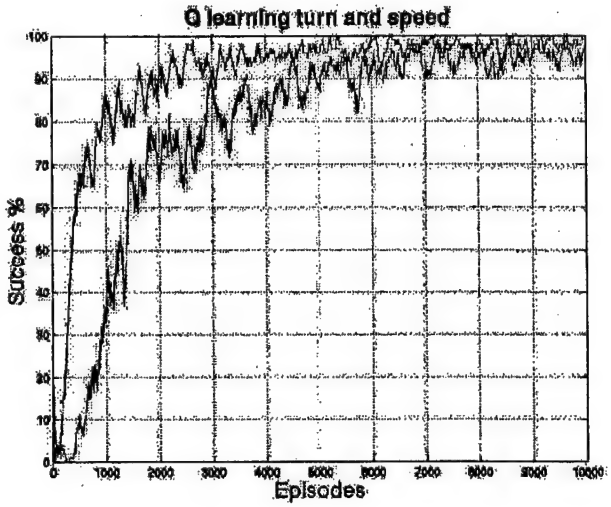


Figure 13: The learning curve of a machine learner using Q learning to acquire the Navigation task. This graph demonstrates that the task of learning turns alone with speed being set automatically by the near-optimal policy is easier than learning speed and turn choices simultaneously.

6.1 Harnessing the results of machine learning

By building a reinforcement learner, we were able to explain characteristics of human learning on the Navigation task. For instance, why does human performance plateau at about 80% while reinforcement learners achieve close to 99% accuracy on the task? Figure 13 reveals that it takes over 10,000 training episodes to move from 80% to 99% competence on the task. Analysis of the reinforcement learner (see Figure 14) reveals an unexpected source of complexity of the task: the most frequently occurring state occurs 45% of the time in an interaction sequence, while states where making appropriate decisions is crucial occur less than 5% of the time. Nearly 10,000 training episodes are needed to get the system to learn the right decisions on these rarely occurring states. It suggests that we can lift human performance to 99% by priming them with these rarely occurring sensor configurations.

The reinforcement learner shows the importance of staged learning for the NRL Navigation task. Learning to make the right turn decisions (while the computer sets the right speed) is an easier problem that learning both turn and speed choices. Once turn decisions are learnt, learning speed choices is simplified for the reinforcement learner. Whether such a staging will help human learning is an open question that needs to be experimentally tested.

A locally non-deceptive intermediate progress function significantly speeds up machine learning. Can we speed up human training protocols by communicating intermediate progress information to our subjects? This is another intriguing possibility raised by our experiments in reinforcement learning.

Preliminary work on a reinforcement learner that acquires the equivalence class structure of the sensor configuration space suggests that with appropriate intermediate progress functions, such partitions can be automatically learned. An avenue for new research is the design of human subject

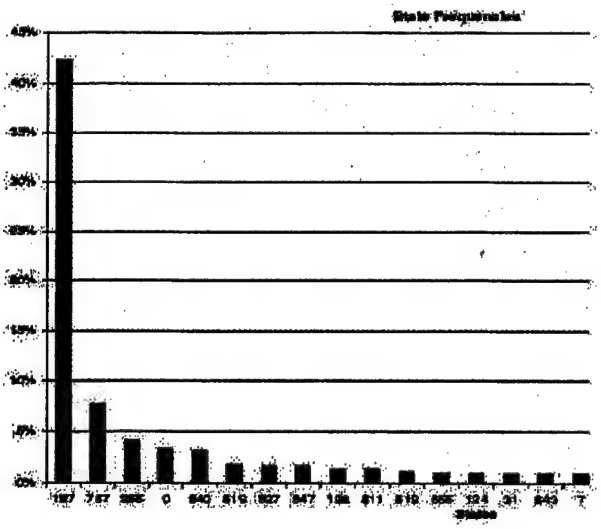


Figure 14: This histogram of states visited by the reinforcement learner shows why it takes over 10,000 episodes to improve performance from 80% to 99% on the NRL Navigation task. We need many many episodes to see the rarely visited states where action choice is critical.

experiments that validate track how humans learn to partition the sensor space and whether that learning process can be accelerated by providing them intermediate progress measures.

7 Conclusions

7.1 Best Accomplishments

Our best accomplishment has been the development of a new real-time computational test to discriminate successful from unsuccessful learners early in the training protocol, for a task with significant strategic and visualmotor components. The test tracks conceptual shifts in action policies constructed from low-level visualmotor performance data gathered during learning, and determines whether or not the subject conforms to the profile of a successful learner. The action policy models that we build from human performance data are detailed enough to capture individual differences and to pinpoint problems in learning, and are simple enough to be built in real-time.

Our hope is that our computational methods will provide a diagnostic tool in the training context for identifying particular learning deficits in this class of tasks. It will lead to a training approach that is custom-fit to each individual using data unobtrusively gathered during task performance. It will discriminate people who fail to learn the task for the lack of an adequate strategy from those who fail to learn the task due to their inabilility to train their visualmotor system to implement a well-designed strategy. It is a scalable solution that harnesses the power of computing to fundamentally change engineering practice in training, and to increase our scientific understanding of human learning.

7.2 Impact

We have developed a modeling technique for a complex visualmotor task of significance to training of submarine pilots. We can now provide individualized training on the basis of our real-time modeling of subject learning derived directly from visualmotor data. This is significant because unlike previous modeling approaches we can extract high level cognitive information viz., strategies and strategy shifts from very low-level objective data from the human visualmotor system. In particular, we can distinguish between learners who fail to acquire a task because of their inability to implement a strategy through their visualmotor system from those that have difficulty formulating a strategy. In real-time, we can pinpoint aspects of the task that a subject is having difficulty learning. We would like to transfer these methods to the Navy training schools for submarine pilots.

7.3 Student Training

This grant supported the training of two graduate students and four undergraduate students at Rice. The graduate students were Sameer Siruguri who completed a graduate thesis on tracking the evolution of learning on the NRL Navigation task, and Raj Bandopadhyay who also completed a graduate thesis. Undergraduates Peggy Fidelman and Scott Griffin built reinforcement learners for the NRL Navigation task and helped elucidate the primary source of complexity in learning the task. Undergraduates Scott Ruthfield and Chris Gouge performed some of the earliest statistical analysis of the visualmotor performance data and paved the way for the machine learning models built by Siruguri.

7.4 Publications

- Tracking the evolution of learning on a visualmotor task Devika Subramanian and Sameer Siruguri, Technical report TR02-401, Department of Computer Science, Rice University, August 2002.
- Tracking the evolution of learning on a visualmotor task Sameer Siruguri, Master's thesis under the supervision of Devika Subramanian, May 2001.
- Inducing hybrid models of learning from visualmotor data , Proceedings of the 22nd Annual Conference of the Cognitive Science Society, Philadelphia, PA, 2000.
- Modeling individual differences on the NRL Navigation task, Proceedings of the 20th Annual Conference of the Cognitive Science Society, Madison, WI, 1998 (with D. Gordon).
- A cognitive model of learning to navigate, Proceedings of the 19th Annual Conference of the Cognitive Science Society, Stanford, CA, 1997 (with D. Gordon).
- Cognitive modeling of action selection learning, Proceedings of the 18th Annual Conference of the Cognitive Science Society, San Diego, 1996 (with D. Gordon)

7.5 Presentations and Invited Lectures

I presented the work in this grant at the following invited lectures and conference presentation.

- Computers and learning, Computer Science Computing and Mentoring Partnership, Rice University, June 2003.
- Distinguished Lucent/CRAW Lecturer, University of Washington, November 2002.
- Invited Speaker, NCARAI Lecture Series at ONR, November 2001.
- Tracking the evolution of learning in a visualmotor task, AI Colloquium series, Texas A&M University, March 2001.
- Tracking the evolution of learning in a visualmotor task. CITI Lunch, December 2000.
- Inducing hybrid models of learning in the NRL Navigation task, Annual Conference on Cognitive Science 2000, Philadelphia, August 2000.
- Progress in learning the NRL Navigation task, ONR workshop, Rice University, August 2000.
- Modeling learning on the NRL Navigation task, invited workshop, Annual Conference on Cognitive Science 1999, Vancouver, August 1999.
- Progress in learning the NRL Navigation task, ONR workshop, San Diego, July 1999 2000.
- Invited Speaker, Workshop on Hybrid Architectures, Cognitive Science, Vancouver, August 1999.
- Distinguished Lecture Series Speaker, Florida Atlantic University, April 1999.
- A study of individual differences in the ONR Navigation task, Annual Conference on Cognitive Science 1998, Madison, Wisconsin, July 1998.
- Learning to Navigate: a new cognitive model, ONR Invited workshop on Hybrid Learning, Corvallis, July 1997.
- A cognitive model of learning to navigate, Annual Conference on Cognitive Science 1997, Stanford, July 1997.
- Invited Speaker, Joint Brazilian Science Foundation and NSF workshop on Intelligent Robotic Agents, March 1997.

References

[1] F. G. Andres and C. Gerloff. Coherence of sequential movements and motor learning. *Journal of Clinical Neurophysiology*, 16(6):520–527, 1999.

[2] C. G. Atkeson, A. W. Moore, and S. Schaal. Locally weighted learning. *Artificial Intelligence Review*, 11:11–73, 1997.

[3] J. Doyon, A. M. Owen, M. Petrides, V. Sziklas, and A. C. Evans. Functional anatomy of visuomotor skill learning in human subjects examined with positron emission tomography. *Eur. J. Neurosci.*, 8(4):637–48, 1996.

[4] D. Gordon and D. Subramanian. Cognitive modeling of action selection learning. In *Proceedings of the 18th Annual Conference of the Cognitive Science Society*, 1996.

[5] D. Gordon and D. Subramanian. A cognitive model of learning to navigate. In *Proceedings of 19th Annual Conference of the Cognitive Science Society*, 1997.

[6] D. Gordon, D. Subramanian, and S. Marshall. Modeling individual differences in the nrl navigation task. In *Proceedings of the 20th Annual Conference of the Cognitive Science Society*, 1998.

[7] S. A. Hillyard and L. Anllo-Vento. Event-related brain potentials in the study of visual-selective attention. *Proc. Natl. Acad. Sci.*, 95:781–787, 1998.

[8] P. Manganotti, C. Gerloff, C. Toro, H. Katsuta, N. Sadato, and P. Zhuang. Task-related coherence and task-related spectral power changes during sequential finger movements. *Electroencephalogr Clin Neurophysiol*, 109:50–62, 1998.

[9] D. E. Meyer and D. E. Kieras. Precis to a practical unified theory of cognition and action: some lessons from epic computational models of human multiple task performance. Technical Report Technical Report, University of Michigan, 1997.

[10] R. C. Miall, G. Z. Reckess, and H. Imamizu. The cerebullum coordinates eye and hand tracking movements. *Nature*, 4(6):638–644, 2001.

[11] A. Newell. *Unified Theories of Cognition.* Harvard University Press, Cambridge, 1990.

[12] S. E. Petersen, H. van Meir, J. A. Fiez, and M. E. Raichle. The effects of practice on the functional anatomy of task performance. *Proceedings of the National Academy of Sciences, U.S.A*, 95(2):853–860, February 1998.

[13] M. Posner, S. E. Petersen, P. T. Fox, and M. E. Raichle. Localization of cognitive operations in the human brain. *Science*, 240:1627–1631, 1988.

[14] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–107, 1986.

[15] E. D. Reichle, P. A. Carpenter, and M. A. Just. The neural bases of strategy and skill in sentence-picture verification. *Cognit. Psychol.*, 40(4):261–295, 2000.

[16] G. L. Schulman, J. M. Olinger, M. Linenweber, S. E. Petersen, and M. Corbetta. Multiple neural correlates of detection in the human brain. *Proceedings of the National Academy of Sciences, U.S.A*, 98(1):313–318, January 2001.

[17] D. Subramanian. Inducing hybrid models of learning from visualmotor data. In *Proceedings of the International Conference on Cognitive Science*, August 2000.

[18] D. Subramanian and S. Siruguri. Tracking the evolution of learning on a visualmotor task. Technical report, Department of Computer Science, Rice University, 2001.

[19] R. Sun and T. Peterson. A hybrid model for learning sequential navigation. In *Proceedings of the International Conference on Robotics and Automation*, pages 234–239, 1997.

[20] R. S. Sutton. Learning to predict by the method of temporal differences. *Machine Learning*, 3:9–44, 1988.

[21] R. S. Sutton. *Reinforcement learning: an introduction.* MIT Press, 1998.

[22] C. J. C. H. Watkins and P. Dayan. Q learning. *Machine Learning*, 8(3):279–292, 1992.